

Digital Design LU

Lab Exercise IV

Jakob Lechner, Thomas Polzer
{lechner, tpolzer}@ecs.tuwien.ac.at
Department of Computer Engineering
University of Technology Vienna

Vienna, November 7, 2011

1 Overview

In this exercise you have to design your own state machines for controlling the 2-line text LCD module which is available on the DE2-115 FPGA board. Designing decent state machines is a key issue for many hardware circuits.

2 Required Reading

- Design flow tutorial
- VHDL modeling slides
- State machine slides
- IP core documentation
- DE2-115 manual
- LCD module manuals

3 Task Description

Task 1: External Interface The LCD is controlled by sending specific commands, which are then executed by the internal controller chip of the display. After power-up the LCD-module first needs to be configured with a certain sequence of commands. Use the commands shown in Listing 1 to set-up the display in 2-line mode. Since the FPGA is connected to all data inputs of the LCD module, you can use the 8-bit interface. This allows you to send a command word (8 bits long) in a single write access.

-
- | | | | |
|----|------------|---|--|
| 1. | "00110000" | — | <i>Function set (8-bit interface)</i> |
| 2. | "00110000" | — | <i>Function set (8-bit interface)</i> |
| 3. | "00110000" | — | <i>Function set (8-bit interface)</i> |
| 4. | "00111100" | — | <i>Function set (8-bit interface, 2 lines, 5x8 font)</i> |
| 5. | "00001111" | — | <i>Display on</i> |
| 6. | "00000001" | — | <i>Display clear</i> |
| 7. | "00000111" | — | <i>cursor moving right, display shift off</i> |
-

Listing 1: Initialization sequence.

Design and implement a state machine (3-process method, Moore FSM), which is able to perform write accesses in order to send commands to the LCD module. Typically the busy flag of the module should be checked before sending

a new command. This, of course, would also require you to implement support for read access on the LCD. For sake of simplicity, you can skip the read out of the busy flag and simply use a timeout mechanism to wait for the execution of the last command to complete before sending a new one. The execution times for all commands can be found in the LCD module manual. After reset this state machine should send the initialization commands given in Listing 1 and then wait for user commands for printing text on the LCD.

Task 2: Internal Interface Section 9 of the IP core manual describes the interface of the video controller, which is used for printing text on the 4.3” touchscreen of the lab boards. Your task for this lab exercise is to design and implement a controller for the 2-line text LCD display, which provides the same interface as the existing video controller. More specifically, this means that the interface signals need to be identical on bit-level (see Listing 17 of IP core manual) and the controller needs to implement the same interface behavior (see Section 9.6 of IP core manual). With an identical interface, the existing video controller and your LCD controller can be used interchangeably.

As presented in the IP core manual the existing controller implements a simple instruction interface (Table 23). Your LCD controller should support all of the specified instructions. For some instructions, however, we introduce minor exceptions from the original behavior, due to the limited capabilities of the 2-line text LCD. Instructions with differing behavior are listed in Table 1.

| Instruction | Description |
|-------------------------------|---|
| <i>INSTR_SET_CHAR</i> | The attributes bits in the instruction data input can be ignored. Furthermore it is not necessary to implement scrolling up the displayed text, if the end of the second line is reached. In this case the cursor can be returned back to the beginning of the first line. Whenever the cursor is moved to the beginning of the next line, clear all characters of this line. |
| <i>INSTR_DELETE</i> | Bits for background color can be ignored. |
| <i>CLEAR_SCREEN</i> | Ignore all parameters of the instruction data input. Simply empty the LCD display. |
| <i>INSTR_MOVE_CURSOR_NEXT</i> | Ignore background color parameter. Whenever the cursor is moved to the beginning of the next line, clear all characters of this line. If the cursor is at the last position of the second line, simply jump to the first position of the first line (no text scrolling). |
| <i>INSTR_NEW_LINE</i> | Ignore background color parameter. Clear all characters of the new line. If the cursor is in the second line, simply jump to the first position of the first line (no text scrolling). |
| <i>INSTR_CFG</i> | No operation. |

Table 1: Adapted LCD controller instructions.

Design and implement a state machine (3-process method, Moore FSM), which processes internal instructions and translates them into one or multiple LCD commands. Use the state machine designed in Task 1 to send these commands to the LCD module.

Task 3: Integration Add the new LCD controller to the top-level design of the project of Lab Exercise III. Connect the *instr*, the *instr_data* and the *wr* inputs to the existing signals, so that the characters produced by the keyboard input will be displayed both on the touchscreen and on the text LCD. Obviously, the *busy* outputs of both display controllers need to be joined by an AND. Synthesize the whole design and download it onto the FPGA board for testing.

4 Submission Specification

The results have to be submitted via myTI. The deadline is November 25th, 2011, 23:59. Upload a tar.gz archive containing the following items:

- Your lab protocol as PDF.
- The complete VHDL source code of the system.
- Your Quartus project (don't forget a cleanup!) or a TCL script creating the project (including the pin mappings!).

Make sure the submitted Quartus project is compilable. All submissions which can not be compiled will be graded with zero points!